Relevance of Log Mining and Analytics Papers to IEEE Transactions on Software Engineering

Domenico Bianculli *University Luxemburg*
Luxemburg
domenico.bianculli@uni.lu
Massimiliano Di Penta *University of Sannio*
Italy
dipenta@unisannio.it
Michael R. Lyu *University of Hong Kong*
Honk Kong, SAR
lyu@cse.cuhk.edu.hk
Sebastian Uchitel *University of Buenos Aires*
Argentina
suchitel@dc.uba.ar
Andy Zaidman *Delft University of Technology*
The Netherlands
A.E.Zaidman@tudelft.nl

Abstract—Within the software engineering field, log analytics aims at extracting and analyzing runtime log data from software systems, with multiple purposes such as detecting anomalies, debugging/locating faults, understanding the system's behavior, or supporting various forms of enhancement, for example, those aimed at improving performance. At the same time, log mining and analytics can also be performed in other contexts, including, for example, the identification of security attacks on a system, or the analysis of a network performance and behavior. The goal of this editorial is to clarify what the IEEE Transactions on Software Engineering sees as relevant contributions in the area of log mining and analytics. Specifically, we provide criteria to determine the relevance of log mining and analytics papers, by also providing examples of what (and what not) would fit the IEEE TSE areas of interest.

Index Terms—Logging Analytics, Logging Parsing, Anomaly Detection

1 Introduction

LOG analytics can be seen as the process of extracting data from (software) systems runtime logs with multiple purposes, including:

- Identifying anomalies in the system's runtime execution;
- Supporting debugging and fault localization;
- Supporting performance analysis and improvement; or
- Understanding the system's (run-time) behavior, therefore aiding its comprehension.

Research contributions in the area of log analytics belong to two main types:

1) **Log mining (also known as "log parsing" or "log message format identification"):** Logs are unstructured (or semi-structured) documents, and conventional parsing techniques for extracting relevant information may not work. Log mining techniques aim to identify the format of the log messages used in logging statements (often called "log templates") through various approaches, e.g., combining heuristics and machine learning models.
2) **Log analytics:** These contributions leverage the information in the log (often, but not necessarily, after having been extracted by log parsing) for the purposes mentioned above.

While there is a long-standing tradition of log mining and analytics in software engineering, the IEEE TSE would like to clarify the extent to which a log analytics paper contributes to the software engineering body of knowledge and, as such, is relevant to the journal. To ensure alignment with TSE's mission of advancing software engineering research, submissions on log mining and analytics should emphasize software engineering relevance, insights into software execution and operation, and integration with software engineering lifecycles.

For what concerns log mining contributions, the relevance to software engineering is relatively straightforward, and they can typically be considered relevant to the software engineering body of knowledge, provided that logs originate from software systems and do not belong to other sources (e.g., network infrastructure or other hardware components).

Concerning log analytics contributions, they could be of different types and do not always belong to software engineering, but may be more relevant to core machine learning, security, or system engineering/dependability. In the following, we first explain what an IEEE TSE log analytics paper should focus on. Then, we provide examples of contributions that are less relevant to software engineering.

1. Focus on Software Engineering Relevance: Contributions should explicitly address software engineering tasks. These can include, for example, fault localization, log-driven testing, or software evolution. Such contributions should explicitly explain how log analytics advances software design, implementation, or maintenance. A possible example could be a piece of research proposing an innovative log-based anomaly detection method *and* investigating some aspects more specific to SE (e.g., the quality or evolution of logging code, the quality/evolution of logs, the anomalous software behavioral patterns, anomalies as violations of some software specification, the link between anomalous logs and source code locations) and their impact on the proposed anomaly detection method. Also, log analytics in the area of DevOps, for example, aimed at analyzing execution logs to aid fault localization, is considered relevant.

2. Provide Insights into Software Execution and Operation. This includes work that addresses software testing and/or operational conditions with the help of log analytics, not just the textual information of the logs. Examples could be pieces of research that are not only able to identify possible errors by analyzing logs but also determine how and why such errors occur (e.g., their root causes). Instead, work that purely applies machine learning techniques as a “black box” to classify system errors without explaining how these errors occurred is not relevant to SE.

3. Map log information with software artifacts, e.g., source code, version control history, or deployment configurations. These include, for example, methods linking log patterns to specific code modules to identify regression bugs or using logs to guide fuzz testing by prioritizing code paths with frequent runtime errors.

Other types of contributions may not properly fit IEEE TSE, but they may be more suitable for other venues. These include:

- Work that mainly contributes to a better machine learning or data mining algorithm without addressing any software system challenge and without explicitly showing any application in the software engineering domain. For example, a better neural architecture that can improve the precision/recall or F1 score of previously proposed general anomaly detection techniques.
- Similarly to the above case, work that provides a novel theoretical contribution (e.g., a new mathematical model for a log stream) without validation in the context of software systems. Examples could be papers proposing a new mathematical representation or graph-clustering method for logs, which are only validated on synthetic datasets.
- Work using logs as “just another dataset” without addressing software-specific challenges (e.g., log-based user behavior analysis unrelated to software and system quality).
- Work performing log analytics on logs completely unrelated to software systems, with no explanation of the detected anomalies/errors, or with analytics that can be more useful in different fields. These include, for example, analytics of networking logs to identify problems in a network infrastructure or anomaly detection to identify security attacks.